DRA: Distributionally Robust Adversarial Training

Nian Si Fan Zhang Teng Zhang

Department of Management Science and Engineering, Stanford University

Abstract

In this paper, we generalize Wasserstein based distributional robust optimization framework to improve the model's robustness against adversarial examples at scale. We develop the defense framework of multi-cost ensemble and adaptive cost learning. We test our method by training a 13 layers neural net on the CIFAR-10 dataset against various white-box attacks. The result shows that the proposed defense achieves the state-of-art performance in ℓ_∞ based attacks and outperforms other methods in ℓ_2 based attacks.

Introduction

Neural network is a powerful model that has excellent explainary power in the domain of vision classifications and speech recognitions, etc. However, it is well known that neural network is vulnerable to adversarial examples. In the context such as automatic driving where mistakes are expensive, the robustness of neural network to adversarial examples received increasing attention from machine learning community.

To remedy the vulnerability to adversarial examples, many methodologies are proposed to generate and defend against the adversarial attacks. In this project, we generalized the Wasserstein based adversarial training procedure proposed by Sinha et al., and test its performance on CIFAR-10 datasets.

Contribution

- We show our distributionally robust adversarial training framework encompasses many other popular heuristic methods used in adversarial training.
- We generalize Wasserstein based adversarial training procedure and test our method in larger data sets (CIFAR10).
- We apply attack ensemble techniques to improve the out-of-sample performance of our defend method.

Model Formulation

The adversarial training procedure can be formulated as a distributionally robust optimization problem

$$\min_{\theta\in\Theta}\sup_{P\in\mathcal{P}} E_P[l(\theta;Z)], \tag{1}$$

where $\mathcal{P}$ is the plausible data distribution set. In this paper, we consider the set as an Wasserstein ball, i.e. $\mathcal{P}=\{P : W_c(P,P_0)\le\delta\}$, where W_c is the Wasserstein metric under cost function c and P_0 is the empirical distribution.

Using stong duality theorem proposed by *Blanchet and Murthy*, (1) is equivalent to

$$\min_{\theta\in\Theta}\inf_{\gamma\ge 0}\gamma\delta + E_{P_0}[\phi_\gamma(\theta;Z)],$$

where

$$\phi_\gamma(\theta;z)=\sup_{\hat{z}}(l(\theta;\hat{z})-\gamma c(z,\hat{z})).$$

To make the training process tractable, we will fix γ and optimize the Lagrangian relaxation

$$\min_{\theta\in\Theta} E_{P_0}[\phi_\gamma(\theta;Z)]. \tag{DRA}$$

Connection with other model: Since the cost function c is free to select, we can recover many common methods in literature using various cost functions. For example, we can recover Basic Iteration Method, Elastic Net Method, *etc.*

Adaptive Cost Function $c(\cdot)$**:** For this method, we consider Mahalanobis distance,

$$c_A(x_1,x_2)=(x_1-x_2)^T A\,(x_1-x_2)$$

where A is chosen by the adversary to produce strongest attack, *i.e.*

$$\min_{\theta\in\Theta}\max_{A\in\mathcal{A}}\mathbb{E}_{\hat{P}_n}\phi_\gamma(\theta;z),$$

Cost Function Ensemble: Suppose we know cost function is ℓ_2 norm with probability p and cost function is ℓ_∞ norm with probability $1-p$. Then, in our training, we randomly pick the cost function with probability p in each steps to train.

Algorithm

```
Input: The empirical distribution $P_0$, learning rate $\alpha$, minibatch size $M$, iteration times $T$.
For $t = 0,\dots,T-1$
    Sample a minibatch from $P_0$ and for each image, find an maximizer $\hat{z}_i{}^t$ of $l(\theta^t;z_i)-\gamma c(z_i,z_i^t)$.
    $\theta_{t+1}\leftarrow\theta_t-\alpha\nabla\frac{1}{M}\sum_{i=1}^M l(\theta^t;\hat{z}_i{}^t))$.
```

Experiments and Result

Dataset We test on the CIFAR-10 data set.

Network structure We test the defenses and attacks based on a 13-layer convolutional neural nets. It consists 10 convolutional layers and 3 fully connected layers intercepted with max pooling layers. This baseline model achieves 79.9% accuracy on the clean test dataset.

Defense models For our distributionally Robust Adversarial Training method, we compare 4 methods respectively: Basic Distributionally Robust Adversarial Training with ℓ_2-norm cost function (DRA_2), Basic Distributionally Robust Adversarial Training with ℓ_∞-norm cost function(DRA_∞), Distributionally Robust Adversarial Training with Multi-cost-function Ensemble ($\text{DRA}_{2-\infty}$) and Adaptive Distributionally Robust Adversarial Training (DRA_A). We compare our models with Fast Gradient Sign Method (FGM), Basic Iterative Method (IGM), Momentum Iterative Method (MGM) under the attack mentioned above.

Result of baseline model under attacks:

FGM_2	FGM_∞	IGM_2	IGM_∞	MGM_2	MGM_∞	DRA_2
0.196	0.107	0.096	0.088	0.097	0.087	0.005

Test accuracy on clean data for various defense model:

Baseline	FGM_2	FGM_∞	IGM_2	IGM_∞	MGM_2	MGM_∞
0.799	0.741	0.739	0.717	0.673	0.713	0.742
	DRA_2	DRA_∞	DRA_A	$\text{DRA}_{2-\infty}$		
	0.652	0.587	0.672	0.577		

Result of ℓ_2 type defense model: Each row corresponds to attacks while each column corresponds to defense models.

	FGM_2	IGM_2	MGM_2	DRA_2	DRA_A
FGM_2	0.439	0.446	0.452	0.490	0.515
FGM_∞	0.155	0.154	0.162	0.206	0.192
IGM_2	0.372	0.397	0.399	0.457	0.492
IGM_∞	0.119	0.121	0.115	0.146	0.141
MGM_2	0.384	0.406	0.407	0.462	0.497
MGM_∞	0.119	0.122	0.116	0.166	0.155
DRA_2	0.071	0.104	0.087	0.270	0.254

Result of ℓ_∞ type defense model: Each row corresponds to attacks while each column corresponds to defense models.

	FGM_∞	IGM_∞	MGM_∞	DRA_∞	DRA_A	$\text{DRA}_{2-\infty}$
FGM_2	0.288	0.517	0.516	0.427	0.515	0.463
FGM_∞	0.645	0.275	0.279	0.226	0.192	0.256
IGM_2	0.139	0.496	0.497	0.407	0.492	0.439
IGM_∞	0.079	0.196	0.201	0.170	0.141	0.201
MGM_2	0.145	0.500	0.500	0.405	0.497	0.440
MGM_∞	0.062	0.196	0.204	0.184	0.155	0.212
DRA_2	0.002	0.247	0.339	0.209	0.254	0.240

Visualization of Adversarial Examples

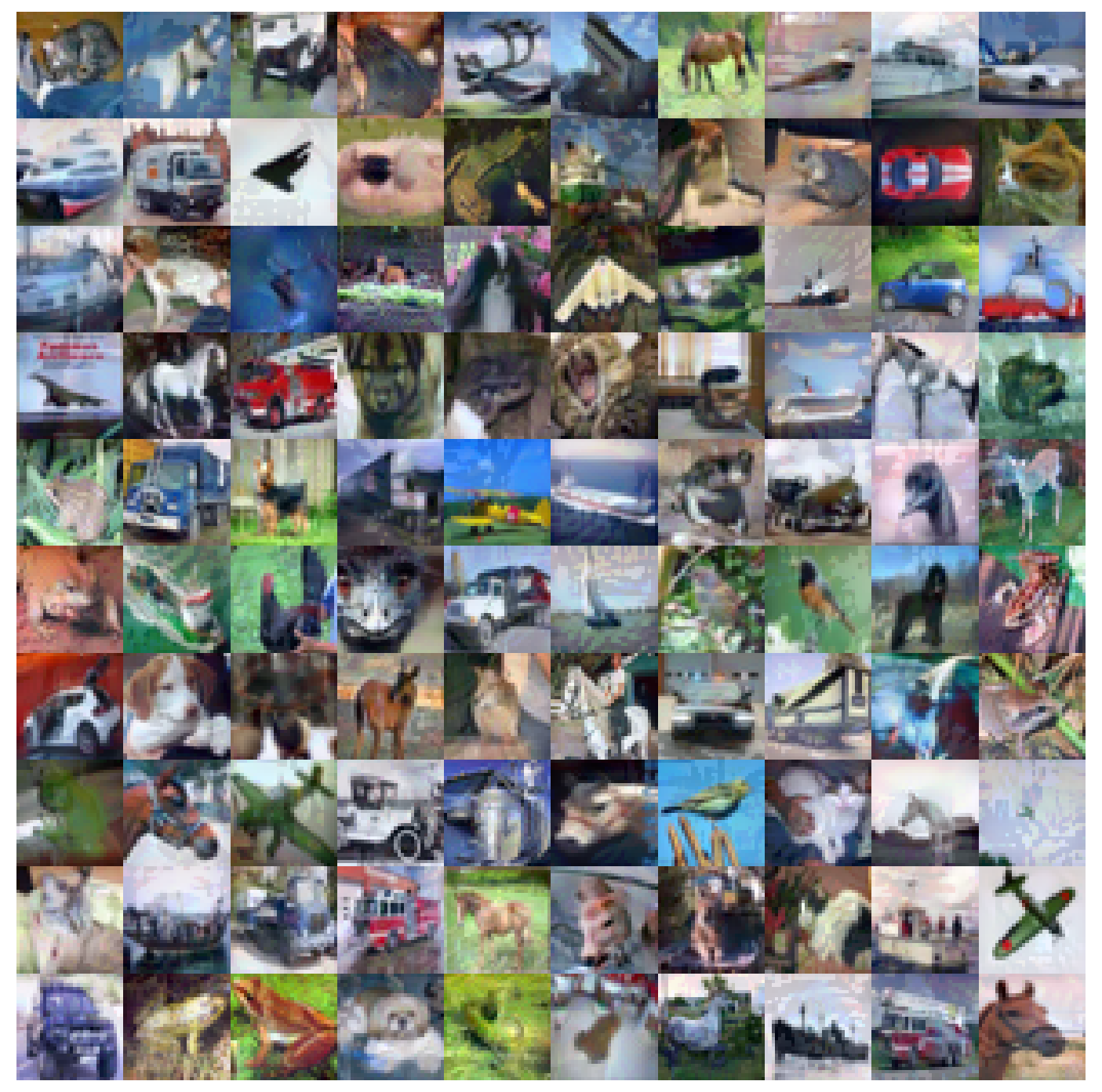

Figure 1: DRA_2

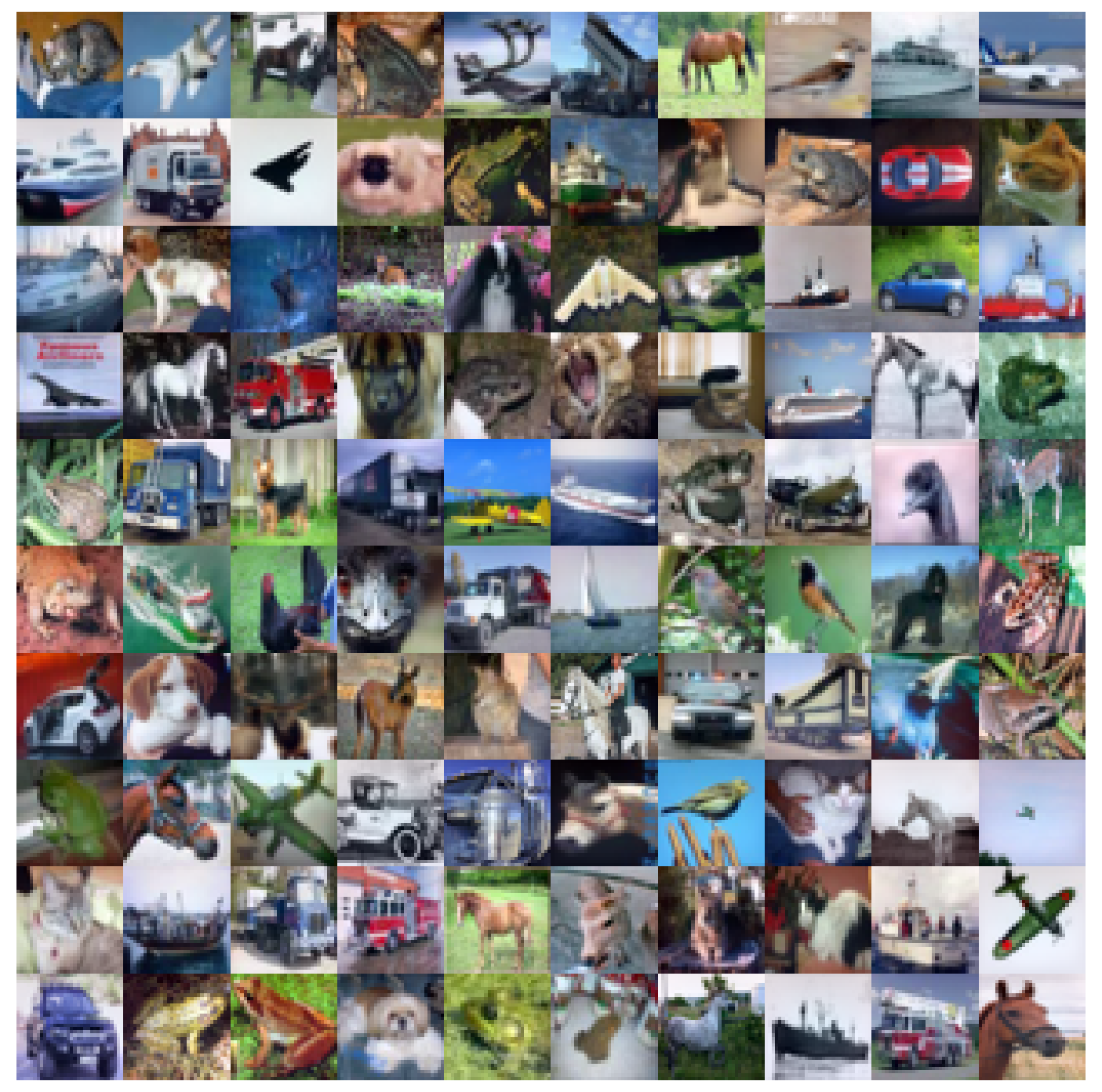

Figure 2: Original Image